AMENDMENT TO THE CLAIMS

This listing of claims will replace all prior versions of claims in the application.

Listing of Claims:

1. (currently amended) A processor comprising:

a first pipeline ~~including a first stage at which instruction results are committed to architected state, wherein the first stage is separated from an issue stage of the first pipeline by a first number of stages~~ having a first number of stages to process instructions;

a second pipeline ~~including a second stage at which an exception is reportable, wherein the second stage is separated from the issue stage of the second pipeline by a second number of stages~~ having a second number of stages, which is greater than the first number of stages, to process floating point instructions; and

a control circuit ~~configured~~ coupled to the first and second pipelines to inhibit co-issuance of a first instruction to the first pipeline ~~and a second instruction to the second pipeline if the first instruction is subsequent to the second instruction in program order~~, if a second instruction to be issued to the second pipeline is a floating point instruction and the first instruction is subsequent to the second instruction in program order, the first instruction to be inhibited for a predetermined number of stage cycles until the first instruction is determined not to graduate from the first pipeline at least no earlier than the second instruction reaching a stage in the second pipeline where exceptions are to be generated to ensure that the first instruction does not graduate from the first pipeline prior to exception determination for the second instruction in the second pipeline.

2. (currently amended) The processor as recited in claim 1 wherein the control circuit ~~is configured to selectively inhibit co-issuance of the first instruction and the second instruction responsive to whether or not exceptions are enabled for the second instruction~~ can disable generation of exceptions.

3. (currently amended) The processor as recited in claim 1 wherein the second

instruction is a long latency floating point instruction ~~and the second pipeline is a floating point pipeline~~.

4. (currently amended) The processor as recited in claim ~~3~~ 1 wherein the first instruction is an integer instruction and the first pipeline is an integer pipeline.

5. (currently amended) The processor as recited in claim ~~3~~ 1 wherein the first instruction is a load/store instruction and the first pipeline is a load/store pipeline.

6. (currently amended) The processor as recited in claim ~~3~~ 1 wherein ~~floating point instructions include short floating point instructions having a first latency during execution,~~ the second instruction is a floating point multiply-add instruction ~~having a second latency during execution which is greater than the first latency, and long latency floating point instructions having a third latency during execution which is greater than the second latency, and wherein the second number of stages is dependent on whether the second instruction is one of the short floating point instructions, the floating point multiply-add instruction, or one of the long latency floating point instruction~~.

7-8. (canceled)

9. (currently amended) The processor as recited in claim ~~8~~ 1 further comprising a scoreboard coupled to the control circuit~~, wherein the control circuit is configured to logically combine the indications in the scoreboard, and wherein the control circuit is configured to permit subsequent issue of instructions responsive to the logical combination having a result indicating that no register writes are pending~~ and the scoreboard includes a set of scoreboard registers to maintain scoreboarding of pending reads and writes.

10. (currently amended) The processor as recited in claim 9 wherein the scoreboard ~~comprises a bit for each~~ uses bits for the scoreboard registers ~~indicative, when set,~~ to indicate that a write is pending ~~to the register, and wherein the logical combination of the~~

~~bits is a logical OR, and wherein the control circuit is configured to permit subsequent issue of instructions responsive to the logical OR being zero~~.

11. (currently amended) A method comprising:

queuing a first instruction for issuance to a first pipeline having a first number of stages;

queuing a second instruction, which is a floating point instruction, for issuance to a second pipeline having a second number of stages, which is greater than the first number of stages, to process floating point instructions;

issuing the second instruction to the second pipeline; and

inhibiting co-issuance of ~~a~~ the first instruction to ~~a~~ the first pipeline~~, and a second instruction to a second pipeline~~ if the first instruction is subsequent to the second instruction in program order, the first instruction to be inhibited for a predetermined number of stage cycles until the first instruction is determined not to graduate from the first pipeline at least no earlier than the second instruction reaching a stage in the second pipeline where exceptions are to be generated to ensure that the first instruction does not graduate from the first pipeline prior to exception determination for the second instruction in the second pipeline~~;~~

~~wherein the first pipeline includes a first stage at which instruction results are committed to architected state, wherein the first stage is separated from an issue stage of the first pipeline by a first number of stages; and wherein the second pipeline includes a second stage at which an exception is reportable, wherein the second stage is separated from the issue stage of the second pipeline by a second number of stages which is greater than the first number~~.

12. (currently amended) The method as recited in claim 11 ~~wherein the inhibiting is selective responsive to whether or not exceptions are enabled for the second instruction~~ further including selective disabling of exceptions in the second pipeline.

13. (currently amended) The method as recited in claim 11 wherein ~~the second instruction is a floating point instruction and the second pipeline is a floating point~~

~~pipeline~~ queuing the second instruction queues a long latency floating point instruction.

14. (currently amended) The method as recited in claim ~~13~~ 11 wherein queuing the first instruction ~~is~~ queues an integer instruction and the first pipeline is an integer pipeline.

15. (currently amended) The method as recited in claim ~~13~~ 11 wherein queuing the first instruction ~~is~~ queues a load/store instruction and the first pipeline is a load/store pipeline.

16. (currently amended) The method as recited in claim ~~13~~ 11 wherein ~~floating point instructions include short floating point instructions having a first latency during execution, a floating point~~ queuing the second instruction queues a multiply-add instruction ~~having a second latency during execution which is greater than the first latency, and long latency floating point instructions having a third latency during execution which is greater than the second latency, and wherein the second number of stages is dependent on whether the second instruction is one of the short floating point instructions, the floating point multiply-add instruction, or one of the long latency floating point instruction, and wherein the method further comprises, if the second instruction is not one of the short floating point instructions, inhibiting co-issuance of subsequent floating point instructions, in program order, to a third pipeline~~.

17-18. (canceled)

19. (currently amended) The method as recited in claim ~~18 wherein the scoreboard comprises a bit for each register indicative, when set, that a write is pending to the register, and wherein the logical combining is a logical ORing, and wherein the permitting is responsive to the logical OR being zero~~ 11 further comprising scoreboarding to maintain pending reads and writes pending in the pipelines.

20. (currently amended) A carrier medium comprising one or more data structures representing a processor, the processor including:

a first pipeline ~~including a first stage at which instruction results are committed to~~

~~architected state, wherein the first stage is separated from an issue stage of the first pipeline by a first number of stages~~ having a first number of stages to process instructions;

a second pipeline ~~including a second stage at which an exception is reportable, wherein the second stage is separated from the issue stage of the second pipeline by a second number of stages~~ having a second number of stages, which is greater than the first number of stages, to process floating point instructions; and

a control circuit ~~configured~~ coupled to the first and second pipelines to inhibit co-issuance of a first instruction to the first pipeline ~~and a second instruction to the second pipeline if the first instruction is subsequent to the second instruction in program order~~, if a second instruction to be issued to the second pipeline is a floating point instruction and the first instruction is subsequent to the second instruction in program order, the first instruction to be inhibited for a predetermined number of stage cycles until the first instruction is determined not to graduate from the first pipeline at least no earlier than the second instruction reaching a stage in the second pipeline where exceptions are to be generated to ensure that the first instruction does not graduate from the first pipeline prior to exception determination for the second instruction in the second pipeline.